Refine Search

Search Results -

Terms	Documents
L20 and L37	18

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L41

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Thursday, December 01, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L41	l20 and l37	18	L41
L40	l37 and l21	0	L40
L39	L38 and l21	14	L39
L38	712/1,205,206,237,238.ccls.	1034	L38
L37	717/149,155,131.ccls.	450	L37
L36	(processor$ near3 state$) near4 (chang$ or modif$ or alter) near7 pipelin$	19	L36
L35	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$ and (trigger and map$ and table$)	5	L35
	DB=TDBD; PLUR=YES; OP=ADJ		
L34	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$	0	L34
	DB=DWPI; PLUR=YES; OP=ADJ		

L33	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$	3	L33
DB=JPAB; PLUR=YES; OP=ADJ			
L32	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$	3	L32
DB=EPAB; PLUR=YES; OP=ADJ			
L31	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$	1	L31
DB=USOC; PLUR=YES; OP=ADJ			
L30	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$	0	L30
DB=PGPB; PLUR=YES; OP=ADJ			
L29	L28 and (conjugat$ near map$ near4 table$)	0	L29
L28	(multi$ or plural$ or first or second$) near4 pipelin$ and pipelin$ near4 (modif$ or chang$ or alter) near5 processor$	31	L28
DB=USPT; PLUR=YES; OP=ADJ			
L27	l23 and (match$ or map$)	1	L27
L26	l23 and (trigger$ same target$)	1	L26
L25	l23 and (ent$ same trigger$ same target$)	0	L25
L24	L23 and pipelin$ near4 (modif$ or chang$ or alter)	1	L24
L23	6304960.pn.	1	L23
L22	L21 and trigger$	14	L22
L21	L20 and (pipelin$ near4 (modif$ or alter or chang$) near9 processor$)	62	L21
L20	L19 and (pipeline$ near5 execu$)	2350	L20
L19	(multi$ or plural$ or first or second$) near4 pipelin$	8934	L19
L18	L17 and (entr$ near4 trigger$ near4 target$)	0	L18
L17	L15 and trigger$	11	L17
L16	L15 and (trigger$ or break$ or halt$) near8 satisf$	0	L16
L15	L14 and (pipelin$ near4 (modif$ or alter or chang$) near9 processor$)	56	L15
L14	L13 and (pipeline$ near8 execu$)	2320	L14
L13	(multi$ or plural$ or frst or second$) near7 pipelin$	8687	L13
L12	(conjugat$ near5 map$ near5 table$)	1	L12
L11	L8 and (map$ or match$ or table$)	1	L11
L10	L8 and (conjugat$ near9 tabl$)	0	L10
L9	L8 and (cnjugat$ near9 tabl$)	0	L9
L8	l1 and trigger$	1	L8
L7	l5 and (trigger$)	5	L7
L6	L5 and trigger$ near9 (satisf$)	0	L6
L5	L4 and (second near4 pipelin$) near9 execut$	24	L5
L4	L3 and (first near4 pipelin$) near4 execut$ near9 (processor$ near4 instruction$)	30	L4
L3	(first near4 pipelin$) and (second$ near4 pipelin$)	3015	L3

L2	L1 and (first$ near4 pipelin$)	1	L2
L1	6035422.pn.	1	L1

END OF SEARCH HISTORY

Terms used
first pipeline and **second pipeline** and **execute** and **processor** and **state** and **conjugate** and **map** and **table**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Compiler-based I/O prefetching for out-of-core applications

Angela Demke Brown, Todd C. Mowry, Orran Krieger

May 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 2

Publisher: ACM Press

Full text available: pdf(499.03 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Current operating systems offer poor performance when a numeric application's working set does not fit in main memory. As a result, programmers who wish to solve "out-of-core" problems efficiently are typically faced with the onerous task of rewriting an application to use explicit I/O operations (e.g., read/write). In this paper, we propose and evaluate a fully automatic technique which liberates the programmer from this task, provides high performance and requires only minima ...

Keywords: compiler optimization, prefetching, virtual memory

2 Exploiting fine-grain thread level parallelism on the MIT multi-ALU processor

Stephen W. Keckler, William J. Dally, Daniel Maskit, Nicholas P. Carter, Andrew Chang, Whay S. Lee

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(1.56 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Much of the improvement in computer performance over the last twenty years has come from faster transistors and architectural advances that increase parallelism. Historically, parallelism has been exploited either at the instruction level with a grain-size of a single instruction or by partitioning applications into coarse threads with grain-sizes of thousands of instructions. Fine-grain threads fill the parallelism gap between these extremes by enabling tasks with run lengths as small as 20 cyc ...

3 Processor-memory coexploration using an architecture description language

Prabhat Mishra, Mahesh Mamidipaka, Nikil Dutt

February 2004 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 3 Issue 1

Results for **"((pipelines and modify and state and processor)<in>metadata)"** e-mail
Your search matched **18** of **1263585** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

((pipelines and modify and state and processor)<in>metadata) >>

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Select Article Information

1. **VLSI implementation of a variable-length pipeline scheme for data-driven**
Yamasaki, T.; Shima, K.; Komori, S.; Takata, H.; Tamura, T.; Asai, F.; Ohno, T Terada, H.;
Solid-State Circuits, IEEE Journal of
Volume 24, Issue 4, Aug. 1989 Page(s):933 - 937
Digital Object Identifier 10.1109/4.34074
AbstractPlus | Full Text: PDF(488 KB) IEEE JNL

2. **Implementation of precise exception in a 5-stage pipeline embedded proc**
Liu Zhenyu; Qi Jiayue;
ASIC, 2003. Proceedings. 5th International Conference on
Volume 1, 21-24 Oct. 2003 Page(s):447 - 451 Vol.1
Digital Object Identifier 10.1109/ICASIC.2003.1277582
AbstractPlus | Full Text: PDF(316 KB) IEEE CNF

3. **Implementing precise interrupts in pipelined processors**
Smith, J.E.; Pleszkun, A.R.;
Computers, IEEE Transactions on
Volume 37, Issue 5, May 1988 Page(s):562 - 573
Digital Object Identifier 10.1109/12.4607
AbstractPlus | Full Text: PDF(1216 KB) IEEE JNL

4. **Razor: a low-power pipeline based on circuit-level timing speculation**
Ernst, D.; Nam Sung Kim; Das, S.; Pant, S.; Rao, R.; Toan Pham; Ziesler, C.; I Austin, T.; Flautner, K.; Mudge, T.;
Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM Inte Symposium on
2003 Page(s):7 - 18
Digital Object Identifier 10.1109/MICRO.2003.1253179
AbstractPlus | Full Text: PDF(686 KB) IEEE CNF

5. **Pipeline LRU block replacement algorithm**
Bhagavathula, R.; Chittoor, P.; Pendse, R.;
Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposiu
Volume 1, 8-11 Aug. 2000 Page(s):404 - 407 vol.1
Digital Object Identifier 10.1109/MWSCAS.2000.951669
AbstractPlus | Full Text: PDF(360 KB) IEEE CNF

Terms used
first pipeline and **second pipeline** and **execute** and **processor** and **state**

Found **101,819** of **167,655**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 On parallel execution of multiple pipelined hash joins

Hui-I Hsiao, Ming-Syan Chen, Philip S. Yu

May 1994 **ACM SIGMOD Record , Proceedings of the 1994 ACM SIGMOD international conference on Management of data SIGMOD '94**, Volume 23 Issue 2

Publisher: ACM Press

Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we study parallel execution of multiple pipelined hash joins. Specifically, we deal with two issues, processor allocation and the use of hash filters, to improve parallel execution of hash joins. We first present a scheme to transform a bushy execution tree to an allocation tree, where each node denotes a pipeline. Then, processors are allocated to the nodes in the allocation tree based on the concept of synchronous execution time such that inner relations (i.e., hash tables) ...

2 Co-synthesis of pipelined structures and instruction reordering constraints for instruction set processors

Ing-Jer Huang

January 2001 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 6 Issue 1

Publisher: ACM Press

Full text available: pdf(1.58 MB) Additional Information: full citation, abstract, references, index terms

This paper presents a hardware/software co-synthesis approach to pipelined ISP (instruction set processor) design. The approach synthesizes the pipeline structure from a given instruction set architecture (behavioral) specification. In addition, it generates a set of reordering constraints that guides the compiler back-end (reorderer) to properly schedule instructions so that possible pipeline hazards are avoided and throughput is improved.Co-synthesis takes place while resolving ...

Keywords: compiler instruction optimization\, instruction set processor, pipeline hazards, pipeline taxonomy, synthesis

3 Pipeline Architecture

C. V. Ramamoorthy, H. F. Li

January 1977 **ACM Computing Surveys (CSUR)**, Volume 9 Issue 1